Induction of Joint Vector-space Embeddings from Heterogeneous Data Sources

Moritz Blum

Bielefeld University, Universitätsstraße 25, Bielefeld, 33615, Germany

Abstract

Recent state-of-the-art approaches in Natural Language Processing and Graph Learning have shown to benefit from the combination of heterogeneous data sources, such as texts and graphs, as these can complement each other in terms of their content. Thus, in order to exploit the available complementary content, complex architectures need to be developed. In this context, semi-structured Knowledge Graphs, which consist of structured data (entities and relations) and unstructured data (literals), are an important information source. However, they are naturally incomplete, and at the same time often contain data of different data types as literals. My thesis proposes to jointly learn vector space embeddings from differently-structured sources to make the data accessible to established approaches without modification of their architecture. It is expected that data sources complement each other, which leads to reduced task training time and increased task performance and robustness. Applications range from Link Prediction in graphs to Named Entity Recognition. My thesis proposes two ways towards a jointly learned embedding space: one solution is to transform all data into a common representation before training, the other possibility is to combine the data during training through data type specific encoders or jointly optimized loss functions.

Keywords

Embedding, Knowledge Representation, Hybrid Machine Learning, Knowledge Graph

1. Introduction: Problem Statement and Importance

Machine Learning (ML) for Natural Language Processing (NLP) or Graph Learning made much progress in the last decades, e. g., in Question Answering [1, 2] and Knowledge Base Completion [3, 4]. However, many approaches lack on task specific data, especially for domain specific applications. One solution is to take background knowledge, e. g., from a Knowledge Graph (KG), into account to make existing data sources usable in all their extent. Some applications already benefit from using data from multiple homogeneous or even heterogeneous data type sources, e. g., Question Answering approaches, which use a KG as background knowledge in addition to a textual database. Consider the following example: A KG contains the information that *Joe Biden* is the *President of the United States* and a text describes that *Jill Biden* and *Joe Biden* are married. Then, both information sources must be combined in order to infer that *Jill Biden* is the *First Lady of the United States.* Such a cognitive disambiguation is natural for humans, but requires additional efforts for machines, e. g., additional preprocessing or a more complex

Doctoral Consortium at ISWC 2022 co-located with 21st International Semantic Web Conference (ISWC 2022)

mblum@techfak.uni-bielefeld.de (M. Blum)
0000-0003-4924-39033 (M. Blum)

CEUR Workshop Proceedings (CEUR-WS.org)

model architecture. Especially, many domain specific datasets complement each other and must be combined, to unleash their full potential.

Data can be distinguished into two broad categories: structured data, which follows a common schema and unstructured data, which does not follow such a schema. Data types of structured data are, e. g., graphs, tables, or vectors, whereas, data types of unstructured data are, e. g., texts or numbers. KGs are denoted as semi-structured, as they contain graph structured data as relational triples, and unstructured data of e. g., textual and numerical representation, as literals in attributive triples. My goal is to develop an approach that allows to combine the information of multiple knowledge representations, where the complementary strength of all can be combined favorably. Instead of separate feature engineering on data sources of heterogeneous data types, this thesis proposes to jointly learn a representation of all information sources in one vector space. This space holds all feature vectors, which aim to capture the semantics present in the sources. Therefore, I hypothise the following benefits of a joint representation:

1. The embedding vectors are task and dataset independent and can be used in most existing Data Mining and ML models without modifications.
2. A smaller amount of task specific training data could be required as data sources complement each other.
3. Increased task performance and robustness.

Furthermore, the approach could be extended to further data types without major changes. Applications are knowledge driven tasks and natural language processing tasks, e. g., Relation Extraction from Text or Entity Disambiguation.

This research can be considered as a contribution towards Neural Symbolic Integration, as a joint embedding space allows the usage of symbolic and subsymbolic data equally. Until now, there are only few approaches which combine multiple data sources of different data types into a common feature space for general application in Data Mining and ML. Semi-structured KGs and a combination of KGs and external textual corpora are promising. Overall, this makes this domain an interesting Ph.D. research topic.

2. Related Work

Vector space embeddings are a common method for feature generation from, e. g., text or graph data, and are usually trained unsupervised on a domain specific task. One prominent example is Word2Vec proposed by Mikolov et al. [5] which learns word embeddings from text. Later, as large language models came out, these were used to generate context sensitive embeddings, e. g., BERT proposed by Devlin et al. [6].

Embedding models were also developed for graph data, too. One method which learns entity embeddings from graphs is RDF2Vec proposed by Ristoski and Paulheim [7]. RDF2Vec applies Word2Vec to graphs by sampling random walks and using them as sentences for training.

Even though there is a lack of methods which treat multiple data types equally, some approaches use additional data to improve the embedding quality or the performance on certain tasks. Such a combination is quite common in the NLP and KG learning domain. Systems like ERNIE [8] proposed by Zhand et al. or CitationIE [9] proposed by Vijay et al. have shown large benefits of using both text and graph data. ERNIE [8] adds entity embeddings of linked entities to an NLP transformer model, whereas CitationIE does a vector concatenation of the input. These are the two most common methods to use multiple input sources in current machine learning approaches, even though of increased dimensionality and redundancy. In the opposite direction, LiteralE proposed by Kristiadi et al. [4] have shown that incorporating numerical features or existing word embeddings into graph embeddings leads to a performance increase in Link Prediction.

Even though past work has shown that vector space embeddings are in general working on different types of data, e. g., Word2Vec and RDF2Vec, research on learning embeddings of different data types in a joint space is less prominent. One example is the approach proposed by Xie et al. [10], which learns structure-based and description-based representations simultaneously in the same vector space in a Link Prediction setting. Their approach relies on a text embedding model to obtain entity representations if textual features are available. Other approaches are simultaneously training text and graph embeddings by defining a joint loss function that is optimized. One prominent example is KEPLER developed by Wang et al. [3] where pretrained language models are jointly optimized regarding a KG and an NLP objective. Another method going into a different direction is EDGE proposed by Rezayi et al. [11]. Their method benefits from augmentation - they augment a Knowledge Graph with external data to learn richer embedding vectors. However, the majority of these methods can only handle two types of data, mostly text and graph data. Furthermore, they are not generally applicable to different domains with different types of data, e. g., numerical data must be treated differently than textual data.

Past work has shown that vector space embedding methods do in general work on different types of data and that the usage of data from multiple sources can lead to performance gains, e. g., in the context of Link Prediction. The existing methods can in principle be used as a starting point to develop methods which learn a joint embedding space. However, none of this models is evaluated in all extend, nor generally applicable to different types of data. Therefore, it is required to outline the possible directions of research and to develop a framework for joint vector space embeddings to increase the quality of vector-space embeddings.

3. Research Questions

The previous section outlines the benefits of a joint embedding space, and present first approaches which combine textual and graph data. The research questions aim to investigate in which way existing methods can be improved and how to develop and evaluate new approaches. A set of downstream tasks which use the embeddings as input will be used to measure their quality of the embeddings. Section 5 provides more details about the evaluation of the embeddings as addressed in the following research questions.

Research Question 1: How do the joint datasets of relational KG triples plus additional data, e. g., literal triples, texts aligned to a KG, or tables aligned to a KG, affect the quality of the joint embedding space? In order to jointly train on multiple data sources, a connection between the datasets is required, s.t. concepts can be related across representations. The methods to create such a dataset will be investigated and compared to each other with respect to the quality of the vector space embeddings and the complexity of the training process.

Research Question 2: The thesis will investigate two methods to create a joint embedding space:

- **Research Question 2.1**: data augmentation - How to transform multiple data sources of heterogeneous type into a common representation, text or graph, for joint training of vector space embeddings of different data types?
- **Research Question 2.2**: joint training - How to combine the encoders and loss-functions of existing feature learning approaches into a single system that learns vector space embeddings jointly on data of different data types?

Research Question 3: What are the effects that arise of the jointly learned vector space embeddings from differently-typed sources, and what are the differences to separated learning? The thesis aims to investigate how the quality of the vector space embeddings for data only contained in one source changes, and how the approaches affect the representation of the data overlap across data sources? Furthermore, this involves the investigation of the approaches concerning the following questions:

- Which effect does data type specific pre-processing and normalization have on the resulting feature vectors, the complexity of the learning process, and the training time?
- How does training complexity and runtime of the developed approaches behave with the amount of data and the number of differently-typed sources?
- Can the approaches combine and use the given data sources in a way such that they overall require less data for training, and even result in high quality vector space embeddings?
- What is the effect of only taking subsets of data, e. g., certain documents, n-hop covers, into account?
- Can the approaches weight data by importance or reliability, and does a weighting have an impact on the learning speed and quality of the embeddings?

4. Preliminary Results

Addressing the research questions, my first work proposes graph transformation as a model independent approach to enable existing Link Prediction models and their embeddings to leverage the literal information in KGs. We hypothesized that the representation of literals as graph structure induces additional information to the learning task, and we have confirmed this hypothesis by the development of literal transformations, which increases the ability of the embeddings for link prediction.

In order to make use of literal data in Link Prediction, we developed transformations to represent these as relational triples, such that existing approaches can leverage this information. We propose the following transformations:

Literal2Entity transforms every literal into an entity, creating a new URI.

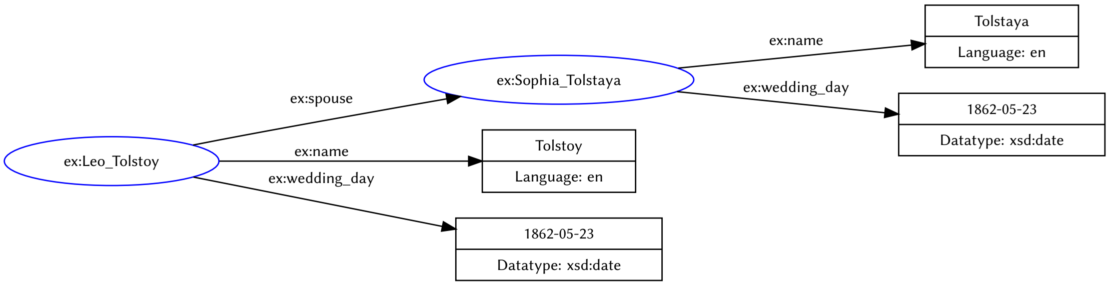

Figure 1: Example: RDF graph that contains literals of type $xsd:date$ and $xsd:string$ with language tag *en*. One can see that $ex:LeoTolstoy$ and $ex:SophiaTolstaya$ share the same wedding day and have a similar name.

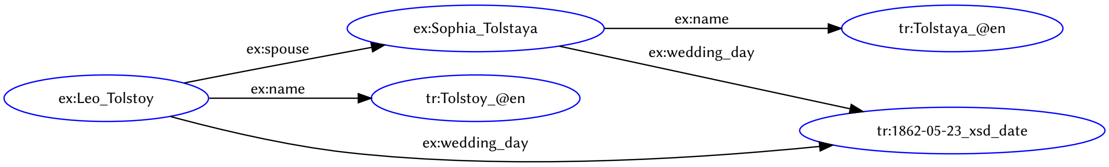

Figure 2: Transformation: *Literal2Entity*, applied to the graph depicted in Fig. 1. This transformation creates a connection between $ex:LeoTolstoy$ and $ex:SophiaTolstaya$ by the shared entity $tr:1862-05-23\_xsd\_date$ because both share exactly the same wedding day.

Datatype2Entity represents the literal's data type as an entity and sets it into relation to the subject entity according to the attributive triple.

Values2Shingles relies on the computation of k-shingles occurring in any textual literal, introducing a URI for each shingle and linking the corresponding subject entity to each of these shingle entities.

Fig. 1 and Fig. 2 show an example RDF KG and the result of the applied *Literal2Entity* transformation.

We compare the performance of established Link Prediction models trained on datasets that are enriched by literal information throughout our transformations against the baseline and against LiteralE. As a baseline, we consider the initial approaches. which do not take literals into account, e. g., DistMult [12] and Complex [13]. In contrast, LiteralE is a framework that describes a modification of certain Link Prediction approaches to let them take literal information into account. In comparison to other methods, which use additional latent literal data representations, we apply a transformation directly on the KG, thus modifying the input, without requiring extensions to the model.

The transformations turned out to beat the baseline and achieve similar performance as LiteralE. The transformations are evaluated on the Link Prediction dataset *FB15k* [14], *FB15k-237* [15], *YAGO3-10* [16], and *LitWD48K* [17]. These datasets are enriched, such that they contain literals of many types and with multiple language tags. For DistMult and ComplEx, the training on the enriched and transformed graphs lead to an up to 11% increased MRR across all transformations. Comparing the scores achieved through our transformations against LiteralE, both approaches show comparable performance. However, our graph transformations integrate the information contained in literals directly into the graph structure and, therefore, allow any existing Link Prediction model to leverage Literal Information.

This experiment shows that combining data in one joint representation leads to improved results in Link Prediction, even though the concrete data value gets missing through the transformation, in some cases. The utility of the embedded literal data is only measured implicitly by their impact to the Link Prediction task, as the data is sparse and can not be used in a specific evaluation schema.

Our approach can be seen as a new baseline to encode literal information into relational triples for Link Prediction. The paper is currently under review at a conference, and we plan to investigate further transformations which are more specifically designed to focus on certain data types, e. g., a transformation with more advanced text operations. The investigation of further transformations is promising, as our first transformations already show state-of-the-art comparable performance.

5. Evaluation

The research questions aim to investigate approaches towards a joint embedding space with respect to the quality of the embeddings. The performance of the developed joint embeddings on these tasks are compared to existing word or graph embeddings. However, the amount of training data and the training time must be taken into account when interpreting the scores.

The evaluation of the developed approaches will be done on existing tasks and datasets from different domains to get reliable results. The applications for evaluation are: Link Prediction on KGs, part-of-speech tagging on texts, and GEval developed by Pellegrino et al. [18] which evaluates feature vectors based on their performance on a set of downstream Machine Learning tasks (classification, regression and clustering) and semantic tasks(entity relatedness and document similarity). Furthermore, the quality of the vector space will be compared by their location on the vectors. There, the neighborhood, the separability of clusters, and the quality of derived analogies, will be compared across the approaches.

The developed approaches will be applied to the DiProMag KG, an RDF KG developed in the context of DiProMag [1] for the description of experiments in the magnetocaloric material science domain. Beyond numerical and textual data, the KG contains data of many data types. In addition, scientific publications are given as background knowledge. Under these conditions, all information must be taken into account equally to derive hypothesis about materials and their experiments. Especially, the literal values contain very valuable information, e. g., about properties of materials. Therefore, a joint embedding space will be learned, and the obtained embeddings will be used for Link Prediction. The derived hypothesis about materials and their properties will be presented to domain experts to rate the quality of the derived information.

[1] https://www.dipromag.de/

6. Conclusion & Future Work

This thesis proposes to represent information from sources of different data types in a jointly learn low dimensional embedding space. The embeddings will work like word or graph embeddings and will be usable as single input to existing Data Mining and Machine Learning approaches. The developed embeddings are facing to capture the semantics present in all representations. First work shows great results, when transforming attributive triples to relational triples, to enrich the training graph for Link Prediction with the information given by literal triples. This work shows the benefits and importance of a joint embedding space.

The next steps are to further investigate the outlined transformation approach. The goal is to design transformations which are specifically designed to focus on certain data types, e. g., a transformation with more advanced text operations. Furthermore, an approach is in development that works on data gathered in a textual representation.

Acknowledgments

I would like to express my special thanks to my supervisors Prof. Dr. Philipp Cimiano and Dr. Basil Ell for all their help and advice. This thesis is written in the context of DiProMag, a BMBF funded research project under Grant No. 13XP5120B.

References

[1] T. Lai, Q. H. Tran, T. Bui, D. Kihara, A gated self-attention memory network for answer selection, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, 2019, pp. 5953–5959.

[2] S. Garg, T. Vu, A. Moschitti, TANDA: Transfer and adapt pre-trained transformer models for answer sentence selection, Proceedings of the AAAI Conference on Artificial Intelligence 34 (2020) 7780–7788.

[3] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, J. Tang, KEPLER: A unified model for knowledge embedding and pre-trained language representation, Transactions of the Association for Computational Linguistics 9 (2021) 176–194.

[4] A. Kristiadi, M. A. Khan, D. Lukovnikov, J. Lehmann, A. Fischer, Incorporating literals into knowledge graph embeddings, in: International Semantic Web Conference, Springer, 2019, pp. 347–363.

[5] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781 (2013).

[6] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2019, pp. 4171–4186.

[7] P. Ristoski, H. Paulheim, RDF2vec: Rdf graph embeddings for data mining, in: International Semantic Web Conference, Springer, 2016, pp. 498–514.

[8] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, Q. Liu, ERNIE: Enhanced language representation with informative entities, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2019, pp. 1441–1451.

[9] V. Viswanathan, G. Neubig, P. Liu, CitationIE: Leveraging the citation graph for scientific information extraction, ACL/IJCNLP (1) (2021) 719–731.

[10] R. Xie, Z. Liu, J. Jia, H. Luan, M. Sun, Representation learning of knowledge graphs with entity descriptions, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 30, 2016.

[11] S. Rezayi, H. Zhao, S. Kim, R. Rossi, N. Lipka, S. Li, EDGE: Enriching knowledge graph embeddings with external text, in: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, 2021, pp. 2767–2776.

[12] B. Yang, W.-t. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, arXiv preprint arXiv:1412.6575 (2014).

[13] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: International conference on machine learning, PMLR, 2016, pp. 2071–2080.

[14] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in neural information processing systems 26 (2013).

[15] K. Toutanova, D. Chen, Observed versus latent features for knowledge base and text inference, in: Proceedings of the 3rd workshop on continuous vector space models and their compositionality, 2015, pp. 57–66.

[16] M. Nickel, V. Tresp, H.-P. Kriegel, Factorizing YAGO: scalable machine learning for linked data, in: Proceedings of the 21st international conference on World Wide Web, 2012, pp. 271–280.

[17] G. A. Gesese, M. Alam, H. Sack, LiterallyWikidata - a benchmark for knowledge graph completion using literals, in: International Semantic Web Conference, Springer, 2021, pp. 511–527.

[18] M. A. Pellegrino, M. Cochez, M. Garofalo, P. Ristoski, A configurable evaluation framework for node embedding techniques, in: The Semantic Web: ESWC 2019 Satellite Events, 2019, pp. 156–160.